Session 6 - Invoking a Subprogram

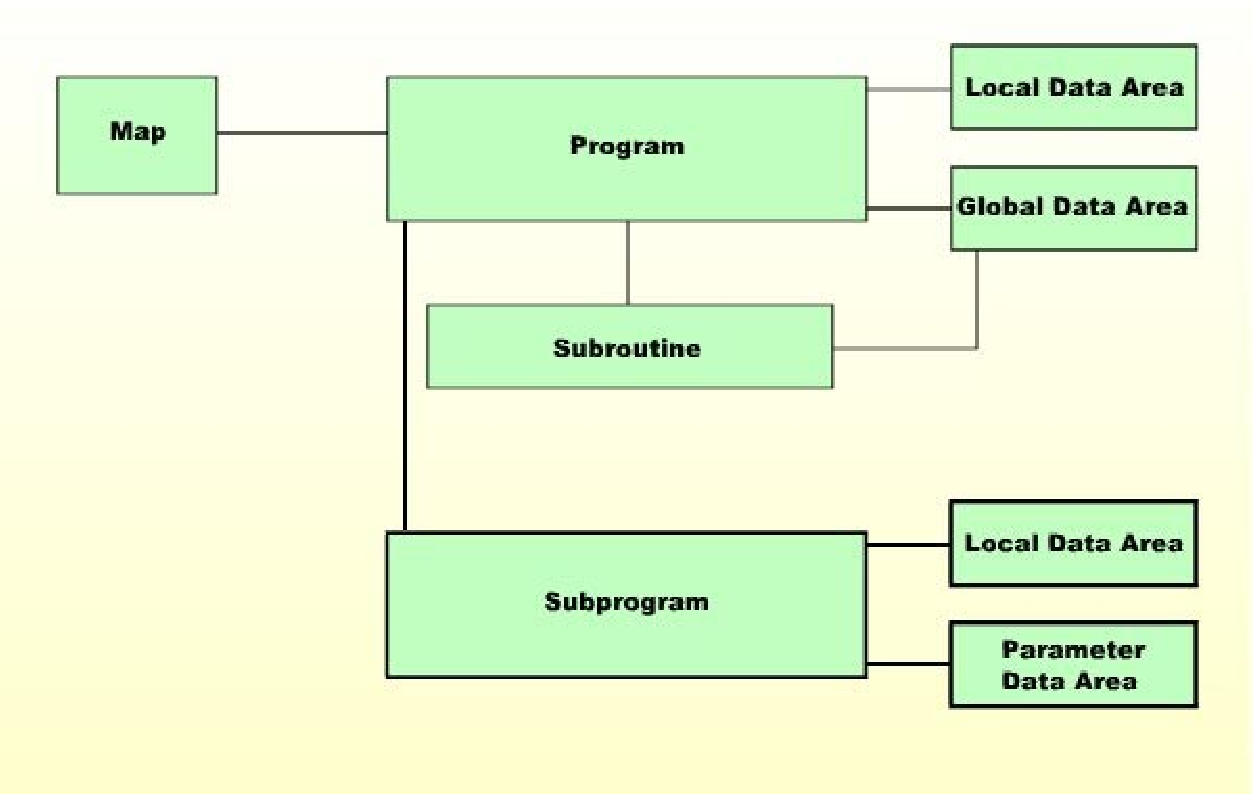
In Natural, both subprograms and subroutines can be invoked from a program. They differ from one another in the way data from the invoking program can be accessed.

As shown in Session 4, a subroutine can access the same global data area as the invoking program. However, a subprogram is invoked with a CALLNAT statement, and with this statement, parameters can be passed from the invoking program to the subprogram. These parameters are the only data available to the subprogram from the invoking program.

The passed parameters must be defined either within the DEFINE DATA PARAMETER statement of the subprogram, or in a parameter data area used by the subprogram.

In addition, a subprogram can have its own local data area, in which the fields to be used within the subprogram are defined.

In this session, you will create a subprogram with a parameter data area and a local data area. In addition, a CALLNAT statement to invoke the subprogram will be added to the main program; also the main program's local data area has to be modified. In the subprogram, the employees selected by the main program will be the basis to select the corresponding vehicle information from the VEHICLES file. As a result, the report will contain vehicle information as well as employees information.

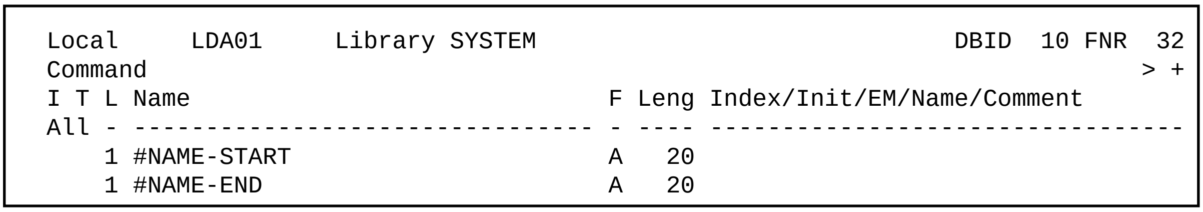


Listed below are Steps 1 to 11 of Session 6.

Step 1

First, the main program’s local data area, LDA01, must be modified.

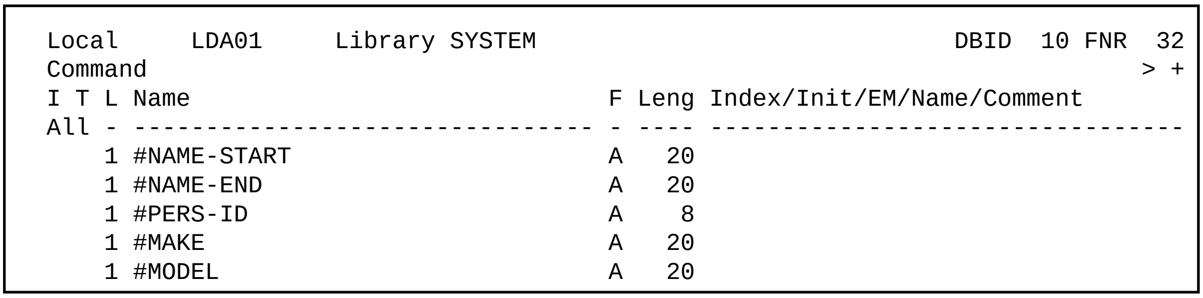
On the Development Functions menu, enter the code **E** (Edit Object) and the name LDA01. The data area editor will be invoked, and the local data area LDA01 will be read into the editing area. LDA01 should appear as follows:



```
Local     LDA01     Library SYSTEM                             DBID  10 FNR  32
Command                                                                     > +
I T L Name                            F Leng Index/Init/EM/Name/Comment
All - -------------------------------- - ---- ---------------------------------
    1 #NAME-START                     A   20
    1 #NAME-END                       A   20
```

Step 2

Add the variables #PERS-ID, #MAKE and #MODEL to the local data area, as show below. They will be referenced in the CALLNAT statement to be added to the program. The local data area now looks as follows:



```
Local     LDA01     Library SYSTEM                             DBID  10 FNR  32
Command                                                                     > +
I T L Name                            F Leng Index/Init/EM/Name/Comment
All - -------------------------------- - ---- ---------------------------------
    1 #NAME-START                     A   20
    1 #NAME-END                       A   20
    1 #PERS-ID                        A    8
    1 #MAKE                           A   20
    1 #MODEL                          A   20
```

CHECK and STOW the local data area.

Step 3

With minor modifications it is possible to use the local data area LDA01 to create the parameter data area that will be needed for the subprogram. (Instead of creating a separate parameter data area, it would also be possible to define the parameter data directly within the subprogram’s DEFINE DATA PARAMETER statement.)

Make a copy of LDA01 by saving it under a different name: at the command prompt of the editor, enter the command SAVE PDA02.

Then enter the command READ PDA02 to read the new copy into the editor.

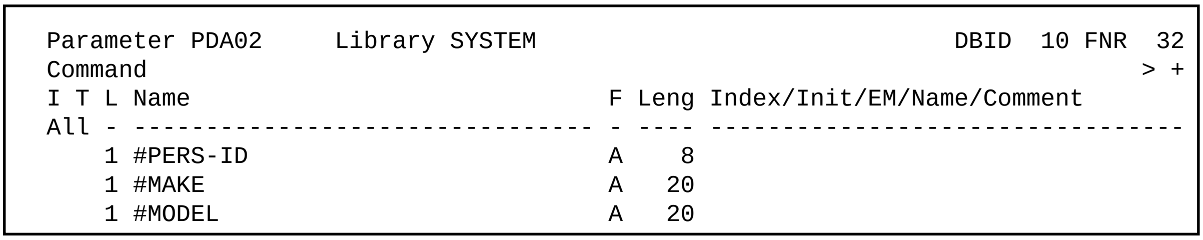
Then enter the command SET TYPE PARAMETER to change the data area type from Local to Parameter.

Step 4

With the line command **.D**, delete the first two lines:

```
1 #NAME-START A 20
1 #NAME-END   A 20
```

The parameter data area now looks as follows:



```
Parameter PDA02     Library SYSTEM                            DBID  10 FNR  32
Command                                                                    > +
I T L Name                            F Leng Index/Init/EM/Name/Comment
All - -------------------------------- - ---- --------------------------------
    1 #PERS-ID                        A    8
    1 #MAKE                           A   20
    1 #MODEL                          A   20
```

CHECK the parameter data area and correct any errors, and then STOW it.

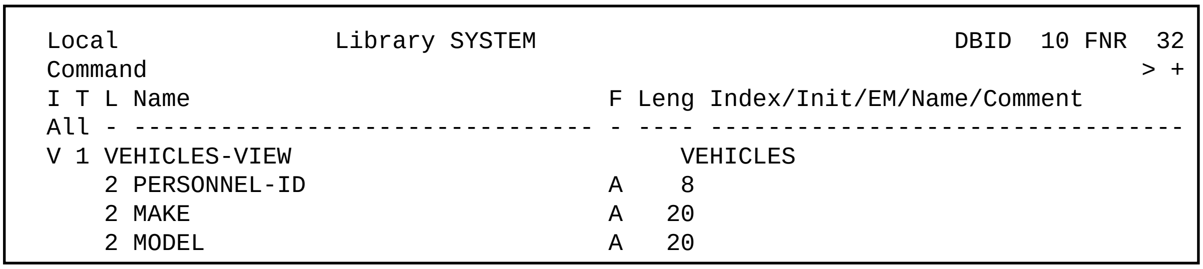
Step 5

Like a program, a subprogram can have its own local data area. This will be created now. First enter the command CLEAR to clear the contents of the editing area. Then change the data area type to Local with the command SET TYPE LOCAL.

Step 6

In the first line of the editing area, starting in the column **T**, enter the line command .V (VEHICLES). The view VEHICLES will be displayed listing all fields contained in that view. Select the following fields to be included into the data area: PERSONNEL-ID, MAKE, and MODEL (see also Session 2, Step 5). These fields will be automatically incorporated into the local data area.

The new local data area should now appear as follows:



```
Local               Library SYSTEM                            DBID  10 FNR  32
Command                                                                    > +
I T L Name                            F Leng Index/Init/EM/Name/Comment
All - -------------------------------- - ---- --------------------------------
V 1 VEHICLES-VIEW                          VEHICLES
    2 PERSONNEL-ID                    A    8
    2 MAKE                            A   20
    2 MODEL                           A   20
```

Step 7

Enter the command SAVE LDA02 to store the data area in source form. Then CHECK the data area, and correct any errors if necessary. Then STOW the data area to compile it and store it in source and object form.

The local data area is now ready to be used by the subprogram.

Step 8

The next step is to create the subprogram itself.

Leave the data area editor by entering a period (**.**) in the Command line. On the Development Functions menu, enter the code **E** (for Edit Object) and the type **N** (for subprogram). The program editor will be invoked with an empty work area.

Type in the subprogram as shown below. Then SAVE it under the name SPGM02 (SAVE SPGM02). CHECK it and correct any errors, and then STOW it.

Subprogram SPGM02:

```
* Example Subprogram ’SPGM02’
* ***************************
DEFINE DATA
  PARAMETER USING PDA02
  LOCAL     USING LDA02
END-DEFINE
*
FD1. FIND (1) VEHICLES-VIEW WITH PERSONNEL-ID = #PERS-ID
       MOVE MAKE (FD1.)    TO  #MAKE
       MOVE MODEL (FD1.)   TO  #MODEL
       ESCAPE BOTTOM
END-FIND
END
```

This subprogram receives the personnel number passed by the main program and uses this number as the basis of a search of the VEHICLES file.

Step 9

Now the main program must be modified to invoke the subprogram.

Read the program into the editor with the command READ PGM01, and insert the following statements immediately before the WRITE TITLE statement:

```
RESET #MAKE #MODEL
CALLNAT ’SPGM02’ PERSONNEL-ID #MAKE #MODEL
```

Some of the parameters passed in the CALLNAT statement are defined in the global data area, and some in the local data area. Note also that the variables defined in the parameter data area of the subprogram need not have the same name as the variables in the CALLNAT statement; it is only necessary that they match in sequence, format, and length.

Step 10

As the program receives vehicle information from the subprogram, the DISPLAY statement must be expanded as follows:

Previous DISPLAY Statement:

```
DISPLAY 23X ’//N A M E’ NAME
         3X ’//DEPT’    DEPT
         3X ’//LV/DUE’  LEAVE-DUE
         3X ’//*’       #MARK
```

Expanded DISPLAY Statement:

```
DISPLAY     ’//N A M E’ NAME
         2X ’//DEPT’    DEPT
         2X ’//LV-/DUE’ LEAVE-DUE
            ’ ’         #MARK
         2X ’//MAKE’    #MAKE
         2X ’//MODEL’   #MODEL
```

After you have made all modifications, the program should look as follows:

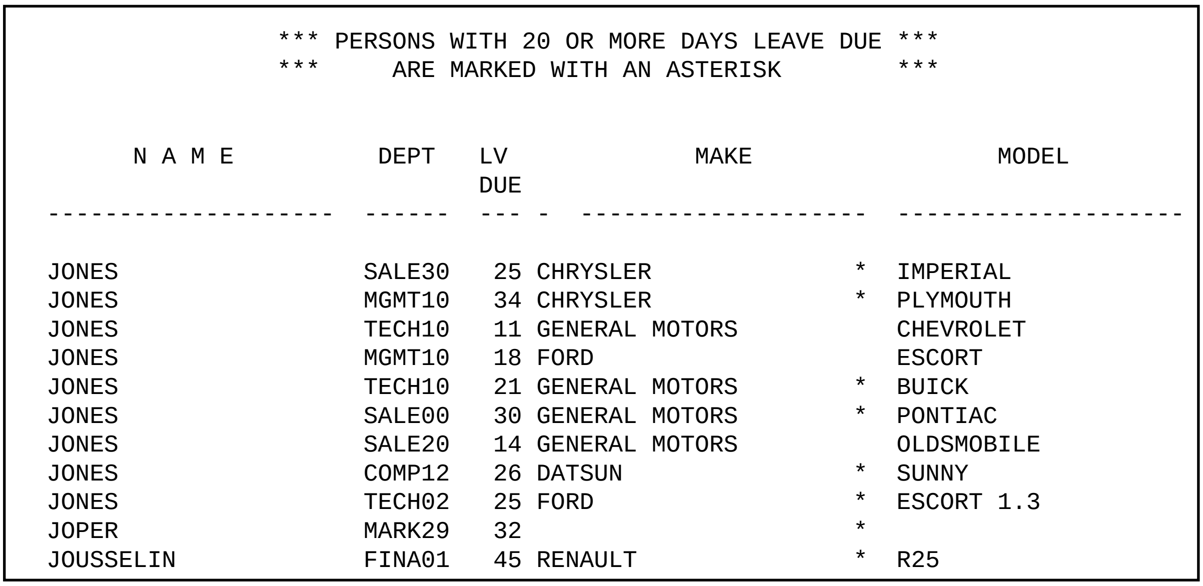
Program PGM01:

```
* Example Program 'PGM01' for Natural Tutorial
* PROGRAM NOW USES A LOCAL DATA AREA
* A GLOBAL DATA AREA AND TITLE HAVE BEEN ADDED AND
*   THE DISPLAY STATEMENT HAS BEEN CHANGED
* THE SUBROUTINE IS NOW EXTERNAL
* A BEGINNING AND ENDING NAME ARE USED FOR THE OUTPUT
* A SUBPROGRAM PROVIDES VEHICLE INFORMATION
* -------------------------------------------------
DEFINE DATA
  GLOBAL USING GDA01
  LOCAL USING LDA01
END-DEFINE
*
REPEAT
*
  INPUT USING MAP 'MAP01'
  IF #NAME-START = '.'
     ESCAPE BOTTOM
  END-IF
  IF #NAME-END = ' '
    MOVE #NAME-START  TO  #NAME-END
  END-IF
*
  RD1. READ EMPLOYEES-VIEW
         BY NAME
         STARTING FROM #NAME-START
         THRU #NAME-END
*
     IF LEAVE-DUE >= 20
       PERFORM MARK-SPECIAL-EMPLOYEES
     ELSE
       RESET #MARK
     END-IF
*
 RESET #MAKE #MODEL
 CALLNAT 'SPGM02' PERSONNEL-ID #MAKE #MODEL                             *
WRITE TITLE / '*** PERSONS WITH 20 OR MORE DAYS LEAVE DUE ***'
            / '***      ARE MARKED WITH AN ASTERISK         ***'//
*
  DISPLAY     '//N A M E' NAME
           2X '//DEPT'    DEPT
           2X '//LV/DUE'  LEAVE-DUE
              ' '         #MARK
           2X '//MAKE'    #MAKE
           2X '//MODEL'   #MODEL                                         *
  END-READ
*
  IF *COUNTER (RD1.) = 0
    REINPUT 'PLEASE TRY ANOTHER NAME'
  END-IF
*
END-REPEAT
END
```

Step 11

Once the program modifications have been made, CHECK the program and correct any errors.

Then RUN the program. On the input screen, enter JONES and JOY as starting and ending names. The report produced by the program now looks as follows:



```
                *** PERSONS WITH 20 OR MORE DAYS LEAVE DUE ***
                ***     ARE MARKED WITH AN ASTERISK        ***


      N A M E          DEPT   LV             MAKE                MODEL
                              DUE
--------------------  ------  --- -  --------------------  --------------------

JONES                 SALE30   25 CHRYSLER              *  IMPERIAL
JONES                 MGMT10   34 CHRYSLER              *  PLYMOUTH
JONES                 TECH10   11 GENERAL MOTORS           CHEVROLET
JONES                 MGMT10   18 FORD                     ESCORT
JONES                 TECH10   21 GENERAL MOTORS        *  BUICK
JONES                 SALE00   30 GENERAL MOTORS        *  PONTIAC
JONES                 SALE20   14 GENERAL MOTORS           OLDSMOBILE
JONES                 COMP12   26 DATSUN                *  SUNNY
JONES                 TECH02   25 FORD                  *  ESCORT 1.3
JOPER                 MARK29   32                       *
JOUSSELIN             FINA01   45 RENAULT               *  R25
```

After the program has been executed, STOW it for future reference.

End of Session 6.